David Groff
CS 192: Programming Service Project
Adviser: Terry Winograd
Organization: Mobilizing Health

Contents

I. Introduction

In this report, I seek to display the results of my independent study through CS 192, Programming Service Project. This document is meant both as a resource for any potential developers of the Mobilizing Health application and as an account of my work for the course.

In the Project Overview, I discuss my goals for the project and explain what I accomplished technically and how this helped the organization. In the documentation overview, I describe where one can go to find more information and where I organized this information in the application. The Notes About Open Source section describes some of my recommendations and requirements for the application to be run as an open source application. I do not justify or explain why initiating an open-source project would be useful. The results section describes the status of Mobilizing Health.

II. Project Overview

My goal was to develop a patient-doctor SMS communication platform that could be run and started on the fly anywhere in India. Prior to my project, an application had been written to run communication between a single set of doctors and patients in Udaipur. After I finished, the program was launched in Pune and West Bengal. See the section on results for more details about the operation. From a technical standpoint, my task involved designing the application to operate in an automated way in places where direct programming intervention had created functionality. For the service aspect, Mobilizing Health wanted to launch the application seamlessly in new areas of India.

In Appendix A, I go over the specific code areas that I implemented as part of this project. The parts described there are the parts that required me to take code that did something specific in the single running instance of the application and generalize the functionality so it could be applied for a new project and customized. In addition to the design component stemming from my own effort, I took notes from potential users in India. I included those notes in Appendix C.

III. Documentation Overview

I commented each piece of the code that required it. In addition, I provide two high-level overviews of the application: a technical introduction to the web application and an overview of the production operation. These files are provided in the doc directory under root. One is README_FOR_APP. The other is called OPERATIONS_OVERVIEW.txt. In addition, I provide a tips list in other.txt. I should say that the documentation here is not meant as a comprehensive course in the application. As in this documentation is not meant to bring someone

who is totally ignorant about the application to a place where they are ready to develop it. I would say that a Rails developer with experience on a similar application would be able to use the documentation in these two READMEs and the comments in the code to quickly get up to speed.

I included two formal documents from Mobilizing Health in Appendix D. These files help to explain the motivation for the model implemented by this application and advertise what results were gained during the operation.

IV. Notes about open source

In this section, I will provide my *opinions* about an open-source effort and what requirements the application needs to function.

A fully functioning Mobilizing Health application as described in this report requires at the very least a web server and a gateway service. A gateway service will be local for whatever country the application serves (if one exists). Some services, such as the popular U.S. service Twilio, have pricing on a per-message basis. At a scale of 150 messages per day or about 10 patients, that service would cost about $50 month. A baseline server costs about $20 per month. The baseline technical requirements cost about $20 a month for a server and up to about $50 a month for an SMS gateway service running in *full production*. In other terms, an application that services 300 patients a month would cost about $70 a month. In addition, the operation often requires providing salaries for field workers and project managers or at least the operation did require such salaries when implemented in Udaipur. Pooja can provide information about her operational payments.

I estimate that an effort to make Mobilizing Health a successful open source application would require three people (or three roles): a front-end developer, a user-experience developer, and a back-end engineer. The real benefit of this application would be the user experience aspect. A lot of the application functionality is already present in Frontline SMS. The real benefit of Mobilizing Health is that it offers two-way communication. Here are some features that would be useful/necessary to create a successful application.

- abstract away the outgoing, or sending, gateway even further so that any SMS service can be “plugged in” (I already factored it into the message model)
- abstract the incoming SMS message request and develop a coherent way to parse incoming SMS messages from different services
- training materials as part of the application i..e, walk-through and help sections
- make the message format extensible: one can choose and add tags such as REQ, HLP, ACC and assign behavior for different tags

V. Results

The service, as provided by the model I worked from, was running live in Udaipur, India from June 2011 to July 2012. In addition, projects spawned in Pune and Salboni after the summer of 2011. Before that, similar solutions had been running in Udaipur as part of Mobilizing Health's effort since July 2010.

In February 2012, Mobilizing Health was acquired by MEDgle Inc. MEDgle offers a service that would analyze the content of incoming text messages, or a message that describes the patient's symptoms. After the analysis, that service would produce an appropriate question. Integrating with Mobilizing Health would have meant that VHDs or patients would send in messages to our system and would go through about three rounds of automated questioning with

MEDgle's system before a message was forwarded to a doctor. As part of the transition, I met with the developers of MEDgle and integrated the above functionality into our system.

MEDgle agreed to maintain the operation developed in India as a non-profit branch of their organization. After about five months of operation, they stopped paying the salaries of workers on the field and thus closed the operation in Udaipur. Since then Mobilizing Health has not been running live. The former CEO and founder Pooja is no longer working full-time on the project. She is studying medicine at UCLA's medical school. She strongly supports any effort to make an open-source project that could be useful for anyone.

Appendix A: Project Milestones and Related Code

Overview

New settings for a project

- Paging schemes (difficulty: normal)
- Paging records
- Notify schemes (difficulty: normal)
- Customize messages (difficulty: easy)
- Provide timing for when messages go out (difficulty: hard)
- VHDs send in messages with an HLP format (difficulty: hard)
- VHDs send in messages with an REQ format (difficulty: easy)
- Messages include doctor's name (difficulty: easy)
- Messages include doctor's mobile (difficulty: easy)
- Select which number to use (difficulty: easy)
- Timezone selection (difficulty: normal)
- VHD requests in a project will page doctor's in that project (difficulty: normal)

Definitions

Project: A hospital or region with 1 set of doctors, Village Health Directors, and Project Managers. Messages from one project's villages typically are not answered by another project's doctors.

VHD (Village Health Director): Village leader who is trained to send SMS text messages on behalf of their community's patients.

PM (Project Manager): The head(s) of the local program.

REQ: Stands for "Request." Every SMS that starts with "REQ" comes from a VHD and indicates that it is a new patient's case. After the REQ comes patient-specific information such as name, age, contact number, and the patient's symptoms.

HLP: Stands for "Help." Every SMS that starts with "HLP" is from a patient whose information (name, age) has already been uploaded into the system and can just start typing their symptoms after the HLP.

Paging Schemes

Code: Paging Schemes (allowing a project to customize the order doctors are paged)

Things to engineer: what order to put doctors in, when a random doctor can be paged, how many doctors to page at once.

How: each project has a list of schemes that specify the priority. the default is for it to pick a random doctor. if the priority has been used for that case, then it is not considered (i.e. once a case uses a scheme with priority 1, it no longer uses any more priority 1 schemes). It uses priority one schemes all at the same time. A scheme can consist of a random doctor so multiple random doctors can be paged at the same time. This occurs if multiple schemes with the same priority are set to page random doctors. That doctor can also be specified.

Test: the simulation will create a project with different paging schemes and simulate rounds of paging.

**There are parameters that determine how many minutes a case pages doctors for and how many minutes it waits before doing the next round of paging which interacts with this code.

Notify Schemes
Code: Notify Schemes (allowing a project to decide what messages are sent to which project managers)
- Implementing notify schemes requires finding all the code that notifies the existing project managers and making it particular to both the project manager and the project. The current system "hard-code" sends a messages to a project manager at random times. So it requires making the notifications more centralized and organized.
Test: simulate setting a project manager to receive a notification (such as new REQ sms) and simulate that action to make sure they would be paged. Then, simulate setting them to not receive the REQ SMS and make sure they are not paged.

Customize Messages
Code: Customize messages (allowing a project to send custom messages instead of the same message) → useful for projects in areas with different languages
- Requires notifications coming out of the database instead of out of hard-code.
Test: Put the customized message in the database, when the action is sending the notification, make sure it is the custom message.

VHDs send in messages with an REQ format
Code: VHDs who bring in patients use REQ format.
- Take the existing code and make it so that this feature can be turned on or off. Also allow it work and route requests properly for the project (as in save for the project and embed whatever logic necessary for saving it under that project).
Test: simulate sending REQ messages into the system.

Messages can include doctor's name/mobile
Code: Messages can include doctor's name/mobile
- If this is turned off, it must format messages to be appropriate. This requires finding the places where messages have that info and having it instead query the database.

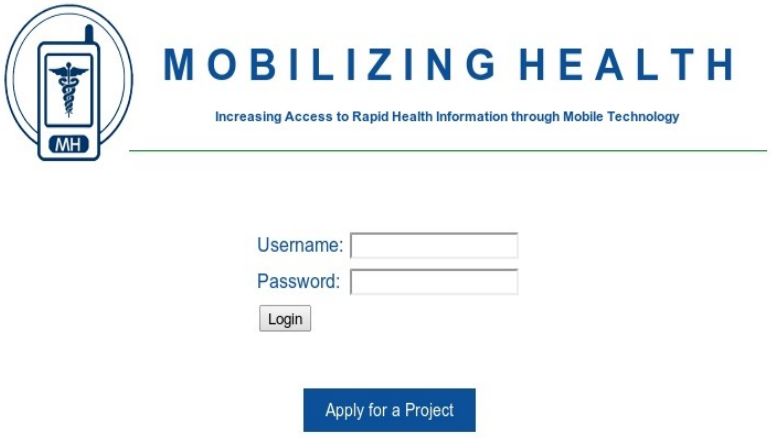
Provide timing for when messages go out
Code: Provide timing for when messages go out (project can set the timings for closing cases that doctors did not answer to, cases that get no response, when PMs are alerted, how often doctors are paged). Write some logic for allowing the timings to work out. The case has to close after PMs are alerted.
- Make this project-specific.
Test: Have to construct data and run the cron. There is no way to run this simulation than by constructing all the pieces of data that go into the calculation.

VHDs send in messages with an HLP format
Code: VHDs who are patients can send in messages with an HLP format.
- Projects can specify that they have patients sending in their own messages and can upload that information. When a message comes in, it must decide if that project allows that format, if the VHD is actually a patient and then actually process the text once it recognizes the HLP format.
- The message must be routed and integrated into the existing paging schemes and format of the REQ. So wherever code gets info out of a REQ, it must be changed to get info out

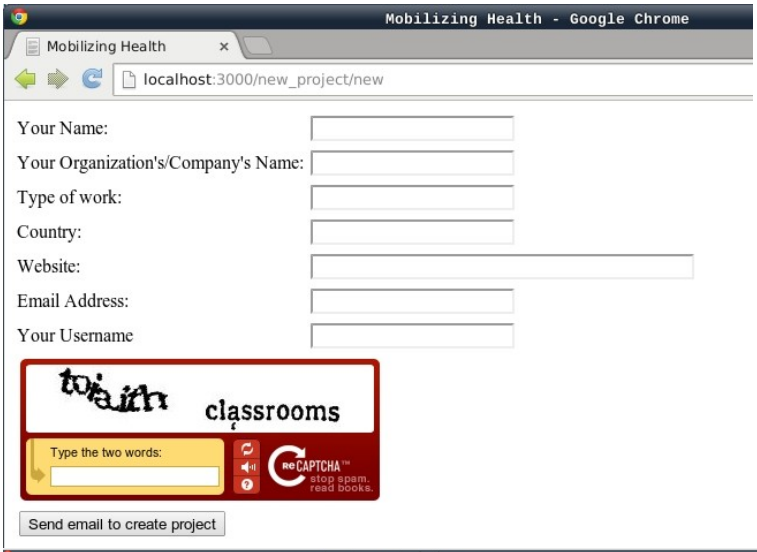
of an HLP or a REQ.
Test: Simulate sending HLP messages into the system and see what messages it sends to doctors and project managers and if it acts properly

Appendix B: Screenshots of Application

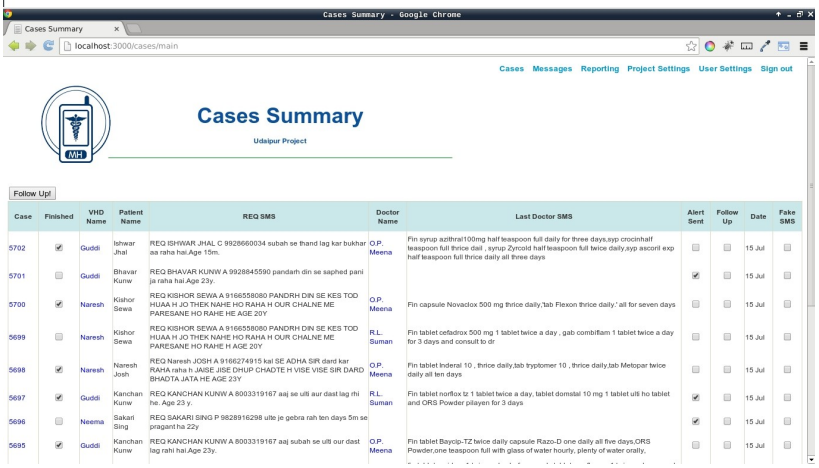
Front-page of application



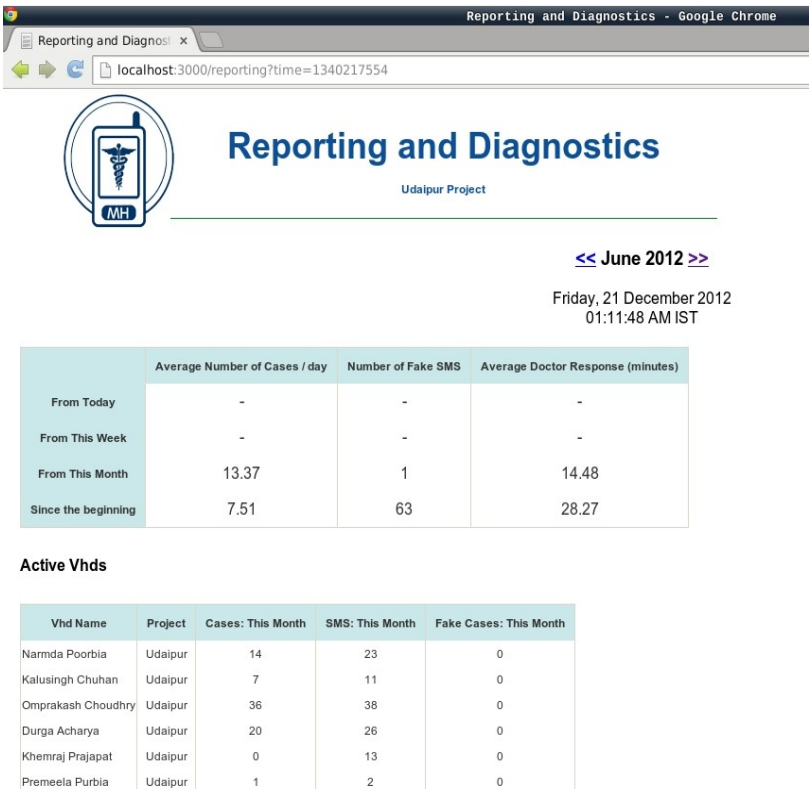
Signing up for a new project through the application → linked from clicking "Apply for a Project" above



Cases summary page that lists all of the most recent cases. The project being viewed is listed under the title. Each column represents a case. The REQ SMS is the message that started the caase.



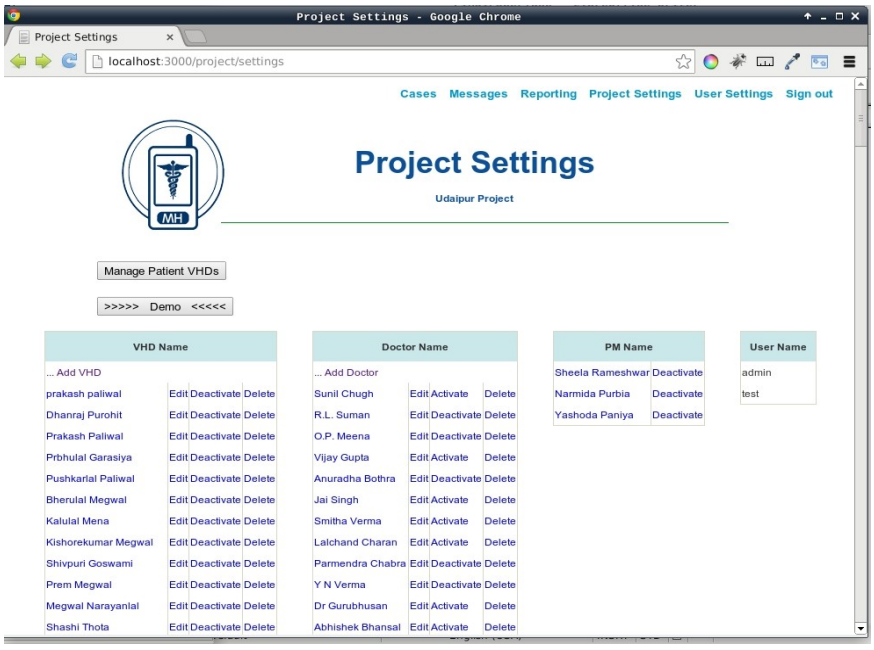
The reporting page that shows some statistics related to the project.



	Average Number of Cases / day	Number of Fake SMS	Average Doctor Response (minutes)
From Today	-	-	-
From This Week	-	-	-
From This Month	13.37	1	14.48
Since the beginning	7.51	63	28.27

Vhd Name	Project	Cases: This Month	SMS: This Month	Fake Cases: This Month
Narmda Poorbia	Udaipur	14	23	0
Kalusingh Chuhan	Udaipur	7	11	0
Omprakash Choudhry	Udaipur	36	38	0
Durga Acharya	Udaipur	20	26	0
Khemraj Prajapat	Udaipur	0	13	0
Premeela Purbia	Udaipur	1	2	0

Here is the project settings page where a project manager can add, delete, deactivate, and edit VHDs and Doctors.



Appendix C: Notes From Users

Notes from Salboni - 21 July 2011
The whole team met with a group of doctors at a government clinic in Salboni, a radical village in West Bengal.

- Concern: we can't charge patients to send SMS because of the structure of the MH service [and because MH would be run as a non-profit service and a not a business service]; most village doctors only make money when they charge patients; quacks could provide only the medicine
- Attractive: doctors like the idea of preventing unnecessary visits and reaching out to patients in the village
- Concern: medico-legal problems and the political situation in West Bengal [a Maoist revolution had occurred in the last five years]
- Proposition: doctors may want to use an "at your own risk" tag or at least note that they are prescribing without seeing the patient
- Attractive: toll-free texting would make the service really attractive

Notes from IIT Kharagpur – 4 August 2011
Pooja and I met with a group of MBA's at IIT Kharagpur including a lead professor.

- Concern: medico-legal problems. Professor thinks that doctors in the Mission Hospital are overly concerned about this as well
- Suggestion: upload the ICD codes for the code dictionary
- Concern: security of the application
- Proposition: they proposed to take over the operation technically. They think the operation is too simple.
- Concern: liability of the doctor. ex: if a patient dies after an SMS; since our system maintains the data [and has to log the data], people may be suspicious that the doctor recommendations in our system led to the death; there is too much evidence
- Problem: don't want to sign any agreement related to our partnership
- Results: they do not want to be a partner; they want a loose agreement and want to work with us very closely

Notes from Miss Sheela – 8 August 2011
Miss Sheela was the project manager in India. At this point, she had been running the operation for a little over a year and had been a major force in designing the initial model of the application. She had seen the application through several iterations.

- Comment: VHDs are more satisfied with recommendations that include medicine; it is not in the doctor's interest to make a thorough treatment
- Suggestion: VHDs should be informed when a message has been marked as fake
- Response: to the doctor game [a way to encourage doctors through rewards]
  - give more points for a scribed case when a doctor takes interest
- Suggestion: doctors login so they can see what they have sent to patients and can send messages to patients and VHDs after they have sent in recommendations

Notes from KEM Vadu – 19 August 2011 and 22 August 2011
KEM Vadu is a hospital in a village of Pune. They undertake a lot of trial projects from different NGOs and businesses worldwide.

- Attractive: codes for VHDs; they provide us with a list of codes that expand for VHDs and we make our system expand those messages
- Concern: they need a lot of funding to start a project
- Comment: there are many unreported diseases in the village that villagers do not want known because it can look bad for the community
- Concern: what texting language do they use
- Comment: VHDs can have varied roles. The ones in Pune at this location were ASHA workers. Should the system store this information or treat VHDs differently?
- Attractive: IVR (aka call-forwarding) through our system so that audio data can be logged online
- Note: KEM Vadu had an SMS system in place that did not work from doctor to doctor (for contacting specialists). Doctors here did not like using their phones for medical-related issues.
- Comment: Demos are necessary for both doctors and patients; they want to be able to train doctors and VHDs in the same room and communicate with each other
- Note: they don't care much for the automatic PHC information. Most people in the village know where the nearest hospitals are.
- Comment: legacy computer systems → Internet Explorer 5; however, the connection speeds are really high
- Comment: they really like the idea of viewing a single patient's history through the application and all the history for a single VHD

Appendix D: Description of Mobilizing Health

I am attaching two PDFs that are materials provided by Mobilizing Health. The first PDF is an executive summary of the organization. Mobilizing Health used this document mostly for marketing purposes. The second PDF is a high-level description of the system. The target audience fort his document is a new partner. It explains the basic usage of the system from a completely external standpoint. By this, I mean the document explains the system from the viewpoint of each user.